VPN の階層的構成法に関する一考察

石橋勇人[1]　山井成良[2]　岡山聖彦[3]　安倍広多[1]　松浦敏雄[1]
[1]大阪市立大学 学術情報総合センター
[2]岡山大学 総合情報処理センター
[3]岡山大学 工学部

概　要

インターネットの発展にともなって VPN の必要性が高まっている．VPN によって，インターネットを通じて遠隔からプライベートな資源にアクセスしたり，インターネットを介して機密性の高い通信を行ったりすることが可能となる．ところが，現在 VPN を実現する手段として主流である IPsec をはじめとして，ほとんどの VPN 実現法では階層的なセキュリティドメイン構成に対応することができなかった．そこで，本稿では，この問題を解決する VPN 構成法を提案する．また，従来方式とのインターオペラビリティを確保する方法についても述べる．

A Method to Establish a VPN Link across Hierarchical Security Domains

Hayato Ishibashi[1], Nariyoshi Yamai[2], Kiyohiko Okayama[3], Kota Abe[1] and Toshio Matsuura[1]
[1]Media Center, Osaka City University
[2]Computer Center, Okayama University
[3]Faculty of Engineering, Okayama University

Abstract

VPN is one of important technologies on the Internet. With VPN, we can access to remote private resource via the Internet. It also makes it possible to send secret messages safely via the Internet. However, there is one significant inconvenience in most current VPN methods like IPsec. Because it is impossible to establish VPN connections across hierarchical security domains. In this paper, we describe a new method that makes it possible to establish VPN connections across hierarchical security domains. It is also inter-operable with former methods by introducing our proxy gateways.

1　はじめに

インターネットの発展にともなって，あらゆる種類の情報がインターネット上を流れるようになってきている．それらの中には，プライバシーに関わる個人情報など秘匿性の高い情報も含まれている．また，地理的に離れた地点間において協調して業務を遂行しようとする場合には，インターネットを利用して通信を行うことによって専用のネットワークを用意するよりも柔軟かつ安価にネットワークを利用できる．

このような場合に使用される技術が VPN (Virtual Private Network) である．VPN を利用することによって，インターネットを介して遠隔地との間に論理的なプライベートネットワークを構築することができ，さらに暗号化技術との組合わせによって安全に情報を伝送することが可能となる．

しかし，IPsec に代表される既存の方式では，VPN が構成できるのは直接 IP 通信が可能な範囲に限定されているため，組織の内部で部分ごとに異なるセキュリティポリシーを持つ場合に，それを反映した階層的なセキュリティドメインを構成すると（直接到達できない）内側のドメインと外部のホストの間では VPN が使用できないという問題があった．そこで，本稿では階層的なセキュリティドメインに対応可能な VPN 構成方式を提案する．また，提案する方式と従来方式との相互接続に関しても考察し，実現法を示す．

2 VPNモデル

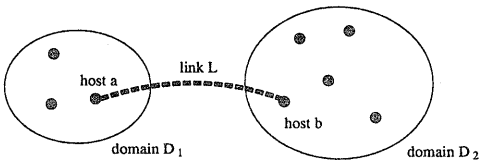

図 1: VPN

VPN は，ネットワーク上の 2 点間に仮想的なリンクを設けることによって，それらがあたかも直接接続されているように見せる技術である．すなわち，ホスト $a \in$ ドメイン D_1 とホスト $b \in$ ドメイン $D_2 (D_1 \neq D_2)$ があるとき，a，b 間に仮想的なリンク L を用意することによって，論理的にはホスト $a \in$ ドメイン D_2（またはホスト $b \in$ ドメイン D_1，以下同様）と見えるようにすることができる（図 1）．

このとき，ホスト a のみを論理的にドメイン D_2 に属させる場合（端末型接続）と，ホスト a を含むドメイン D_1 のホスト全体をドメイン D_2 に属させる場合（LAN 間接続）がある．後者は前者においてホスト a をルータとして動作させることによって実現可能なので，以下では前者の場合について論じることにする．

ところで，一般に VPN において通信内容の秘密を守ることは必須の要件ではないが，実際の局面では暗号化等による通信内容の秘匿化や認証の機能が必要とされることが多い．本稿では，これらのセキュリティに関する機能を含んだ VPN 機能の実現を考える．そこで，アクセスポリシーを基準としてドメインを規定し，同一のアクセスポリシーを持つ範囲をセキュリティドメインと呼ぶことにする．

あるセキュリティドメインとその外部を結び付ける接点には，セキュリティゲートウェイ（SGW）が設置される．逆に言うと，SGW を経由せずに通信できる範囲が同一のセキュリティドメインということになる．SGW はセキュリティポリシーが異なるドメインとの間の通信を制御する．

3 セキュリティドメインの構成

大学や企業などの組織におけるもっとも単純なセキュリティポリシーの与え方は，組織内部のどこでも一様なポリシーを与えることである．この場合，組織全体が 1 つのセキュリティドメインとなる．

ところが，ある程度以上の規模を持つ組織では，情報へのアクセス可能性に関して内部が一様ではないことが多い．すなわち，一般に組織内の特定のセクションのみがアクセス可能な情報が存在する．また，組織内部の他のセクションからはアクセスできないが，外部の特定の組織からはアクセス可能な情報も考えられる．たとえば，大学のネットワークに接続された附属病院のデータを，他の学部等からはアクセスできないようにしつつ外部の病院からはアクセスできるようにする場合や，企業の中にある研究所が外部の研究機関と共同研究を行う場合などである．前者の例では大学全体が 1 つのセキュリティドメインであり，附属病院もまた 1 つのセキュリティドメインとなる．

このように，1 つの組織の中に複数のセキュリティドメインが構成されることは一般的で，しかもそれは階層的に構成される（図 2）．

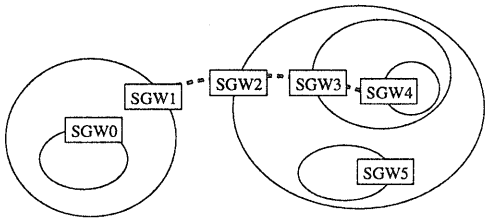

図 2: 階層的なドメイン

4 既存の VPN 実現法とその問題点

IP ネットワーク上で VPN を実現するために使用可能な方法には，L2TP（Layer 2 Tunneling Protocol）[1]，IPsec（IP Security Protocol）[2, 3]，VTun（Virtual Tunnel）[4]，SSH（Secure Shell）[5]，SOCKSv5[6]，SSL（Secure Sockets Layer），TLS（Transport Layer Security）[7] 等様々なものがある（表 1）．

これらのうち，利用者から意識することなく，すなわち，個別のアプリケーションへの組み込みやサービスごとの設定を必要とせずに利用できるのはネットワーク層以下のレベルでデータを伝送する方式（IPsec，L2TP 等）である．

これらの方式はいずれも VPN を終端するネットワーク機器同士が IP レベルで直接通信可能であることを要求する．しかし，本稿で対象とするような

表 1: 様々なプロトコル

	暗号化あり	暗号化なし
アプリケーション層	SSH（遠隔ログイン）	
セション層		SOCKS
トランスポート層	SSL, TLS, SSH*（*ポートフォワーディング）	
ネットワーク層	IPsec, VTun	
データリンク層		L2TP

階層的セキュリティドメイン構成では，VPN を終端させたい SGW と外部のホストが直接通信することができないため，従来方式では VPN を構成することができないという問題がある．

また，IPsec のようにヘッダを認証する機能がある場合，現行のネットワークで広く普及している NAT（Network Address Translator）のようなヘッダを変換する方式との両立は困難である[1]．

5 提案する方式

本章では階層的なセキュリティドメインを構成した上で VPN が利用できるように，まず仮想パスを形成してそれを利用する方式を提案する．

5.1 満たすべき条件

セキュリティドメインが階層構成をなすようなネットワークにおいて VPN による安全な通信を行おうとする場合，満たすべき条件としては次のようなものがある．

1. VPN 上で利用できるアプリケーションやトランスポートプロトコルを限定しない
2. 暗号化による通信内容の保護ができる
3. オーバーヘッドが少ない
4. 階層的なセキュリティドメイン構成に対応できる
   （経路上に SGW が複数存在しても使用できる）
5. 各 SGW ごとに独立して管理が行える
   （ユーザ認証方式の決定，ユーザ管理など）
6. SGW を認証することができ，安全に VPN を構成できる
7. NAT に対応できる

また，接続のために既存の VPN クライアントソフトウェアが使用できるならば，なお良い．

[1]ヘッダの認証をしないなど，いくつかの制約条件を付加することによって可能な場合もある [8].

5.2 方式の検討

階層的なセキュリティドメインの内部に含まれる SGW に対して VPN リンクを張ろうとする場合，直接通信可能なのは隣り合う SGW の間だけであるから，何らかの形で 1 レベルずつ順に階層を降りて行く必要がある．

このような場合の VPN リンクの構成法として

1. VPN リンクの生成を要求するホスト（イニシエータ）から，経路上で通過する SGW に対して VPN リンクを張ることを 1 ホップずつ繰り返して VPN リンクを伸ばしてゆく
2. 隣接 SGW 間にそれぞれ VPN リンクを張って結合する
3. イニシエータと最終的な目的の SGW の間に仮想パスを作り，その中に VPN リンクを構成する

が考えられる．

1. は，VPN を要求しているイニシエータと，もっとも外側の SGW との間で一旦 VPN を構成し，次に 1 レベル内側の SGW とイニシエータの間で VPN を構成することを再帰的に繰り返す方式である．この場合，VPN リンクが通過する SGW の数だけ重畳して形成され（図 3），オーバーヘッドが大きくなる．特に，暗号化を行う際にはこのオーバーヘッドが顕著なものとなる．

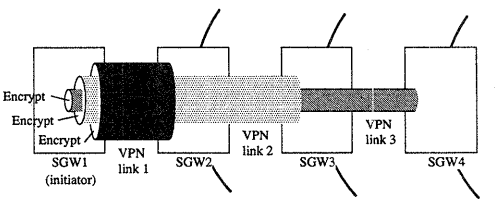

図 3: 方式 1

2. は，隣り合う SGW 間でそれぞれ VPN リンクを構成し，それらをつなぎ合わせる方式である（図 4）．この場合，暗号化を考えると，各 SGW において復号化と暗号化のプロセスを繰り返すことになるため，やはりオーバーヘッドが大きいという問題がある．また，各 SGW において内容が一度復号化

されるため，終端以外の点において秘密が漏れる可能性がある．

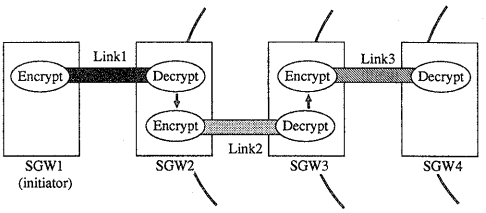

図 4: 方式 2

3. は，イニシエータと最終的な目的の SGW の間に仮想的なパス（ポイント・ポイントリンク）を用意し，そのパスの中に VPN を構成する方式である（図 5）．この場合には，暗号化はエンド・エンドで行われるため，上の 2 つの方式のように余分なオーバーヘッドがかかることはない．したがって，この方式がもっとも効率的でかつ安全である．そこで，我々の提案ではこの方式を採用する．

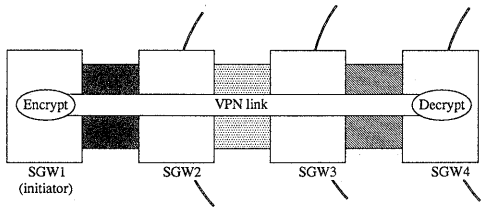

図 5: 方式 3

次に，3. の方式を実現するために我々が提案するプロトコル SPSP（Simple Path Signalling Protocol）の動作の概略を示す．

5.3 動作の概要

イニシエータと目的の SGW との間で上で述べた仮想パスを確立することは，次のような手順で実現できる．

1. イニシエータは，相手ドメインの入り口ゲートウェイ SGW（SGW_{in}）に対してコネクションを張る．SGW_{in} の IP アドレスはあらかじめ与えておくものとする．
2. コネクションが確立すると，イニシエータは SGW_{in} に対して，トンネルを要求したユーザ名ならびに接続しようとする SGW（SGW_{dest}）を通知する．
3. SGW_{in} はイニシエータに対して，必要ならば認証要求を行う（ポリシーによっては行わなくても良い）．
4. 認証に成功するか，あるいは認証が必要ない場合，SGW_{in} は仮想パス用ルーティングテーブルを参照し，次ホップとなる SGW へコネクションを張る．
5. 次ホップへのコネクションを張った SGW は，以降パケットを透過的に転送する．
6. 隣り合う SGW を順にたどりながら，必要なだけ 2~5 と同様な処理を繰り返す．
7. SGW_{dest} に達すると，コネクション確立メッセージが 1 つ手前の SGW に返され，これが繰り返されて順にイニシエータまで遡る．
8. これによって仮想パスが形成されるので，エンドシステム同士で直接 IP 通信が可能となり，IPsec などを用いて暗号化メッセージをやり取りすることができるようになる．

上のような手順を用いて形成した仮想パスの中に IPsec 等を用いた VPN を構成することによって，ネットワーク層レベルでの暗号化された接続が可能となり，5.1 の条件 1 および 2 を満たすことができる．5.2 で述べたように，本方式ではオーバーヘッドは少なく，条件 3 を満たしている．手順から明らかなように条件 4 を満たすことができ，認証のための情報はそれを必要とする SGW のみが知っておけば良いため，条件 5 を満たす．イニシエータ及び各 SGW 間のやり取りは，X.509 証明書や公開鍵暗号方式を用いて認証することによって相手を確認できるので，条件 6 も満たすことが可能である．また，通過するドメインにおいて NAT が使用されている場合であっても，直接通信するのは隣り合う階層の SGW 同士であり，さらに IPsec は論理的に直結されたパス上においてエンド・エンド間で適用されるために問題はない．したがって条件 7 も満たすことができる．

このように，提案する方式では上の手順によって 5.1 の各条件をすべて満足させることができる．

6 実装に関する検討

ここでは，前章の方式を実装するにあたって考慮すべき点について検討する．

6.1 SGW のアドレッシング

5.3 の手順 2 において，SGW_{dest} を通知するために使用する識別子は，ドメイン内で SGW が一意に識別できるならば何でも良い．ただし，階層性を持つ識別子を用いることによって，セキュリティドメインの階層構造を識別子に反映することが可能となり，仮想パスのルーティング処理を簡略化できるという利点がある．たとえば，FQDN (Fully Qualified Domain Name) や IPv6 アドレスがこれにあたる．

階層性は別として，識別子として IPv4 アドレスを使用することも可能ではある．ただし，その場合には，すべての SGW_{dest} がグローバルな IP アドレスを持つ必要がある．このためには，(1) LAN で使用しているネットワークアドレスがグローバルなものである，または，(2) 静的な NAT によって各 SGW のアドレスをそれぞれを異なるグローバルな IP アドレスに 1 対 1 に変換する，のいずれかの条件を満たすことが要求される．

6.2 仮想パスのルーティング

SGW 間で仮想パスを形成するために，仮想パスをルーティングするためのテーブルが必要である．これは，SGW_{dest} が与えられた場合に次ホップとなる SGW を求めるために使用する．

ところで，6.1 において，宛先の SGW の識別子の代表的な例として FQDN ならびに IPv6 アドレスをあげた．これらを使用した場合には，ルーティングを IP 層に任せることによって仮想パスのルーティングテーブルを省略することも可能である．なぜなら，目的の IP アドレスがわかっていればそこへコネクションを張ることが可能であり，また，FQDN がわかっている場合にはネームサーバを利用して IP アドレスを取得した上でコネクションが張れるからである．ただし，この場合には，SGW_{dest} への IP から見た経路と仮想パスを構成するための経路が一致していることが条件となる．

逆に，仮想パス用のルーティングテーブルを使用することによって，IP の経路とは独立して SGW_{dest} へのトンネルを構成することができる．この場合には，ルーティング時にユーザの情報を利用することによって，ユーザごとに異なるゲートウェイを使用することも可能である．

6.3 ルーティング情報の与え方

前節で述べた仮想パス専用のルーティングテーブルを使用する場合，これを作成するためには，大きく分けて 2 つの方法が考えられる．静的に与える方法と，動的に作成する方法である．

前者の場合には管理者があらかじめ各 SGW に定義しておけば良いので簡便であるが，SGW の追加や削除，故障による一時的な経路変更等の際に管理者がテーブルを再定義する必要があり，管理が煩雑となる．

後者の場合には，ルーティングプロトコルを使用して SGW 間で互いにルーティング情報を交換することになる．この場合には自動的にルーティングテーブルを更新できるため，経路の変更に対しては柔軟に対応できる．ルーティングプロトコルとしては RIP や OSPF など既存の IP 用のルーティングプロトコルをベースとしても良いが，(1) 階層的に上下関係が定まっている，(2) 同一階層における SGW の数は一般にそれほど多くない，(3) SGW は互いに異なるブロードキャストドメインに属することが多い，(4) 認証が重要である，などの理由により，1 つ上位の SGW に対してルーティング情報を伝達するプロトコルを定めて使用することにする．この際，X.509 証明書あるいは公開鍵暗号方式によって SGW を認証する．

7 既存方式との整合性

本稿で提案するような新しい方式を導入する場合，当然イニシエータ側もそのプロトコルに対応する必要がある．これは，ネットワーク接続型 VPN の場合には比較的対応が容易であるが，端末接続側 VPN の場合には，イニシエータとなる端末は多数存在し，かつ利用者の手元にあるために導入が困難な場合がある．そこで，本章では，イニシエータ側には変更を行うことなしに利用可能とするための代理ゲートウェイ (proxy gateway) 方式について述べる．

代理ゲートウェイとは，IPsec 等従来のプロトコルによるイニシエータからの VPN 接続要求を SPSP によるシグナリングに変換するゲートウェイである．この場合，イニシエータ側は従来通りのプロトコルを用いているため，最終的に目的とする SGW を指定する機能を特に持っていない．したがって，SGW の指定はパケットヘッダの宛先 IP アドレスによって行うことになる．このため，SGW がグローバル

にアドレッシング可能であることが必要であり，特にIPv4の場合には6.1で述べたのと同じ条件を満たさなければならない．

代理ゲートウェイは概略次のようにして実現できる．

1. イニシエータは目的のゲートウェイ（SGW_{dest}）に対して通常どおり直接VPNリンクを張ろうとする．
2. SGW_{dest}への経路上に存在する入り口ゲートウェイSGW_{in}は，これを検知し，SPSPを用いてSGW_{dest}へ向けて仮想パスのシグナリングを行う．
3. この際，特殊なフラグを立てておくことによって，経路途中のSGWはイニシエータに対する認証要求を行わないようにする．
4. SGW_{in}からSGW_{dest}までの仮想パスが確立すると，SGW_{in}はシグナリングの契機となったパケットを仮想パスへ送出する．これ以降のパケットも同様に仮想パスへ中継することにより，VPNリンクが成立する．

この場合，イニシエータ側が使用するVPNプロトコルは階層を考慮しない既存のものであるため，各SGWごとに個別に認証を行うことができない．このため，代理サービスであることを示すフラグによって経路上のSGWにおける認証をスキップさせ，SGW_{dest}のみが認証動作を行うようにする．認証に失敗すると，SGW_{dest}からコネクション切断メッセージが送信され，経路上の各SGWは仮想パスを切断する．

代理ゲートウェイ方式では，ユーザの認証が最終的な目的ゲートウェイであるSGW_{dest}のみで行われるため，途中のSGWはユーザ認証なしに仮想パスの生成を受け入れる必要がある（ユーザ認証なしでは仮想パスの接続を許可しないSGWが経路上に存在した場合，そこから先へは接続できない）．すなわち，利用ポリシーはすべてSGW_{dest}が制御することになるため，各階層のポリシーが反映できないという点で柔軟性に欠けるという欠点があるものの，既存のVPNクライアントをそのまま収容できるという点では非常に有効である．

8 おわりに

本稿では，階層的なセキュリティドメイン構成に対応可能なVPN実現方式を提案し，実装のために必要となる要素について検討した．また，IPsecのような従来方式と本方式との相互接続を可能とするための代理ゲートウェイについてもその実現法を示した．これによって，従来は困難であった部門ごとのセキュリティポリシーを柔軟に実現することが可能となる．今後の課題としては，実際に本方式を実装して評価を行うことがあげられる．また，提案した方式の機能の一部はSOCKS5[9]の多段プロキシー機能によっても実現可能と考えられるので，比較して評価を行いたい．

参考文献

[1] Pall, G. S., Palter, B., Rubens, A., Townsley, W. M., Valencia, A. J. and Zorn, G.: Layer Two Tunneling Protocol "L2TP", RFC 2661 (1999).

[2] Kent, S. and Atkinson, R.: Security Architecture for the Internet Protocol, RFC 2401 (1998).

[3] Thayer, R., Doraswamy, N. and Glenn, R.: IP Security Document Roadmap, RFC 2411 (1988).

[4] Krasnyansky, M.: Virtual Tunnels over TCP/IP networks, http://vtun.sourceforge.net/.

[5] SSH Communications Security: SSH Secure Shell, http://www.ssh.org/.

[6] Leech, M., Ganis, M., Lee, Y.-D., Kuris, R., Koblas, D. and Jones, L.: SOCKS Protocol Version 5, RFC 1928 (1996).

[7] Dierks, T. and Allen, C.: The TLS Protocol Version 1.0, RFC 2246 (1999).

[8] Aboba, B.: NAT and IPSEC, Internet Draft (2000). draft-aboba-nat-ipsec-01.txt.

[9] NEC: Welcome To The SOCKS5 Framework Site, http://www.socks5.nec.com/socks5_index.html.